Generators and Diffie–Hellman

250H

Generators

- Let p be a prime and $g \in \{1, . . . , p-1\}$. *g is a generator for mod p* if

$$\{g^1, . . . , g^{p-1}\} = \{1, . . . , p-1\}.$$

Generators

- Let p be a prime and $g \in \{1, . . . , p-1\}$. *g is a generator for mod p* if

$$\{g^1, . . . , g^{p-1}\} = \{1, . . . , p-1\}.$$

- This set $\{1, . . . , p-1\}$ on multiplication is known as $\mathbf{Z}_p^*$

Generators

- Let p be a prime and $g \in \{1, . . . , p-1\}$. *g is a generator for mod p* if

$$\{g^1, . . . , g^{p-1}\} = \{1, . . . , p-1\}.$$

- This set $\{1, . . . , p-1\}$ on multiplication is known as $\mathbf{Z}_p^*$
- Note that $g^{k*l} = g^{l*k}$ for $k, l \in \mathbf{Z}$

Example of generators for $\mathbf{Z}_p^*$

- Consider $\mathbf{Z}_2^*$
  - $\mathbf{Z}_2^* = \{1\}$
  - What is the generator for $\mathbf{Z}_2^*$?

Example of generators for $\mathbf{Z}_p^*$

- Consider $\mathbf{Z}_2^*$
  - $\mathbf{Z}_2^* = \{1\}$
  - What is the generator for $\mathbf{Z}_2^*$?
    - 1

Example of generators for $\mathbf{Z}_p^*$

- Consider $\mathbf{Z}_2^*$
  - $\mathbf{Z}_2^* = \{1\}$
  - What is the generator for $\mathbf{Z}_2^*$?
    - 1
- Consider $\mathbf{Z}_3^*$
  - $\mathbf{Z}_3^* = \{1, 2\}$
  - What is the generator for $\mathbf{Z}_3^*$?

Example of generators for $\mathbf{Z}_p^*$

- Consider $\mathbf{Z}_2^*$
  - $\mathbf{Z}_2^* = \{1\}$
  - What is the generator for $\mathbf{Z}_2^*$?
    - 1
- Consider $\mathbf{Z}_3^*$
  - $\mathbf{Z}_3^* = \{1, 2\}$
  - What is the generator for $\mathbf{Z}_3^*$?
    - 1^n will just give us back 1 so 1 can't be a generator
    - $2^1 = 2$

      $2^2 = 1$

      So 2 is a generator for $\mathbf{Z}_3^*$

Consider $\mathbf{Z}_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

- Generators for $\mathbf{Z}_{11}^*$
  - 2: [2, 4, 8, 5, 10, 9, 7, 3, 6, 1]
  - 6: [6, 3, 7, 9, 10, 5, 8, 4, 2, 1]
  - 7: [7, 5, 2, 3, 10, 4, 6, 9, 8, 1]
  - 8: [8, 9, 6, 4, 10, 3, 2, 5, 7, 1]

Consider $\mathbf{Z}_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

- Generators for $\mathbf{Z}_{11}^*$
  - 2: [2, 4, 8, 5, 10, 9, 7, 3, 6, 1]
  - 6: [6, 3, 7, 9, 10, 5, 8, 4, 2, 1]
  - 7: [7, 5, 2, 3, 10, 4, 6, 9, 8, 1]
  - 8: [8, 9, 6, 4, 10, 3, 2, 5, 7, 1]
- Numbers that are not generators for $\mathbf{Z}_{11}^*$
  - 1: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
  - 3: [3, 9, 5, 4, 1, 3, 9, 5, 4, 1]
  - 4: [4, 5, 9, 3, 1, 4, 5, 9, 3, 1]’
  - 5: [5, 3, 4, 9, 1, 5, 3, 4, 9, 1]
  - 9: [9, 4, 3, 5, 1, 9, 4, 3, 5, 1]
  - 10: [10, 1, 10, 1, 10, 1, 10, 1, 10, 1]

The Discrete Logarithm Problem

- For any integer b and primitive root a of prime number p, we can find a unique exponent i such that

$$z \equiv g^i \pmod{p} \text{ where } 0 \leq i \leq (p-1)$$

The Discrete Logarithm Problem

- For any integer b and primitive root a of prime number p, we can find a unique exponent i such that

$$z \equiv g^i \pmod{p} \text{ where } 0 \leq i \leq (p-1)$$

- Do you think it is difficult for a computer to find i?

The Discrete Logarithm Problem

- For any integer b and primitive root a of prime number p, we can find a unique exponent i such that

$$z \equiv g^i \pmod{p} \text{ where } 0 \leq i \leq (p-1)$$

- There is no efficient classical algorithm known for computing discrete logarithms in general

What is Easy and What is Hard for a Computer

- Easy
  - Powers: a^b mod p
  - Finding a prime p and a generator g for Z_p* (we have not done this, but it's true)

What is Easy and What is Hard for a Computer

- Easy
  - Powers: a^b mod p
  - Finding a prime p and a generator g for Z_p* (we have not done this, but it's true)

- HARD:
  - Discrete Log (Actually a close cousin of DL, but we won't get into that.)

Diffie–Hellman Key Exchange

- The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for encryption of messages

Diffie–Hellman Key Exchange

- The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for encryption of messages
- It allows a way in which a public channel can be used to create a confidential shared key

Diffie–Hellman Key Exchange

- The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for encryption of messages
- It allows a way in which a public channel can be used to create a confidential shared key
- The algorithm is depends on the difficulty of a problem similar to computing discrete logarithms
  - Given g^a and g^b, find g^{ab}

Diffie-Hellman

1. Alice and Bob publicly agree on a large prime p to be what we are modding by
   Alice and Bob publicly agree on a generator for mod p: g

Diffie-Hellman

1. Alice and Bob publicly agree on a large prime p to be what we are modding by
   Alice and Bob publicly agree on a generator for mod p: g
2. Alice selects a secret key: a
   Bob selects a secret key: b

Diffie-Hellman

1. Alice and Bob publicly agree on a large prime *p* to be what we are modding by
   Alice and Bob publicly agree on a generator for mod *p*: *g*
2. Alice selects a secret key: *a*
   Bob selects a secret key: *b*
3. Alice combines her secret key *a* with the generator and prime that were decided on: $A = g^a \bmod p$
   Bob combines his secret key b with the generator and prime that were decided on: $B = g^b \bmod p$

Diffie-Hellman

1. Alice and Bob publicly agree on a large prime *p* to be what we are modding by
   Alice and Bob publicly agree on a generator for mod *p*: *g*
2. Alice selects a secret key: *a*
   Bob selects a secret key: *b*
3. Alice combines her secret key *a* with the generator and prime that were decided on: $A = g^a \bmod p$
   Bob combines his secret key b with the generator and prime that were decided on: $B = g^b \bmod p$
4. Alice and Bob share their values with each other

Diffie-Hellman

1. Alice and Bob publicly agree on a large prime *p* to be what we are modding by
   Alice and Bob publicly agree on a generator for mod *p*: *g*
2. Alice selects a secret key: *a*
   Bob selects a secret key: *b*
3. Alice combines her secret key *a* with the generator and prime that were decided on: $A = g^a \bmod p$
   Bob combines his secret key b with the generator and prime that were decided on: $B = g^b \bmod p$
4. Alice and Bob share their values with each other
5. Alice computes: $z = (B \bmod p)^a \bmod p$
   Bob computes: $z = (A \bmod p)^a \bmod p$

Diffie-Hellman

1. Alice and Bob publicly agree on a large prime *p* to be what we are modding by
   Alice and Bob publicly agree on a generator for mod *p*: *g*
2. Alice selects a secret key: *a*
   Bob selects a secret key: *b*
3. Alice combines her secret key *a* with the generator and prime that were decided on: $A = g^a \bmod p$
   Bob combines his secret key b with the generator and prime that were decided on: $B = g^b \bmod p$
4. Alice and Bob share their values with each other
5. Alice computes: $z = (B \bmod p)^a \bmod p$
   Bob computes: $z = (A \bmod p)^a \bmod p$
6. z is the shared secret key that can be used to encrypt and decrypt messages to each other

Diffie-Hellman Example

1. Alice and Bob publicly agree p = 353
   Alice and Bob publicly agree g = 3

Diffie-Hellman Example

1. Alice and Bob publicly agree p = 353
   Alice and Bob publicly agree g = 3
2. Alice selects a secret key: 97
   Bob selects a secret key: 233

Diffie-Hellman Example

1. Alice and Bob publicly agree p = 353
   Alice and Bob publicly agree g = 3
2. Alice selects a secret key: 97
   Bob selects a secret key: 233
3. Alice finds A: 3^{97} mod 353 = 40
   Bob finds B: 3^{233} mod 353 = 248

Diffie-Hellman Example

1. Alice and Bob publicly agree p = 353
   Alice and Bob publicly agree g = 3
2. Alice selects a secret key: 97
   Bob selects a secret key: 233
3. Alice finds A: $3^{97} \bmod 353 = 40$
   Bob finds B: $3^{233} \bmod 353 = 248$
4. Alice and Bob share their values with each other

Diffie-Hellman Example

1. Alice and Bob publicly agree p = 353
   Alice and Bob publicly agree g = 3
2. Alice selects a secret key: 97
   Bob selects a secret key: 233
3. Alice finds A: $3^{97} \bmod 353 = 40$
   Bob finds B: $3^{233} \bmod 353 = 248$
4. Alice and Bob share their values with each other
5. Alice computes: $z = (248 \bmod 353)^{97} \bmod 353 = 160$
   Bob computes: $z = (40 \bmod 353)^{233} \bmod 353 = 160$

Diffie-Hellman Example

1. Alice and Bob publicly agree p = 353
   Alice and Bob publicly agree g = 3
2. Alice selects a secret key: 97
   Bob selects a secret key: 233
3. Alice finds A: 3^{97} mod 353 = 40
   Bob finds B: 3^{233} mod 353 = 248
4. Alice and Bob share their values with each other
5. Alice computes: z = $(248 \bmod 353)^{97}$ mod 353 = 160
   Bob computes: z = $(40 \bmod 353)^{233}$ mod 353 = 160
6. 160 is the shared secret key

Diffie-Hellman

- In the example on the previous slide, it would be possible to use brute force to find 160

Diffie-Hellman

- In the example on the previous slide, it would be possible to use brute force to find 160
- In particular, Eve could determine the common key by finding the solution to $3^a \bmod 353 = 40$ or $3^b \bmod 353 = 248$

Diffie-Hellman

- In the example on the previous slide, it would be possible to use brute force to find 160
- In particular, Eve could determine the common key by finding the solution to 3^a mod 353 = 40 or 3^b mod 353 = 248
- Eve could just calculate the powers of 3 mod 353 and stop when she gets to 40 or 248

Diffie-Hellman

- In the example on the previous slide, it would be possible to use brute force to find 160
- In particular, Eve could determine the common key by finding the solution to $3^a \bmod 353 = 40$ or $3^b \bmod 353 = 248$
- Eve could just calculate the powers of 3 mod 353 and stop when she gets to 40 or 248
- With large numbers brute force becomes impractical

Diffie-Hellman

- From -500BC to 1976AD if two people wanted to set up a crypto system they had to meet to exchange keys.

Diffie-Hellman

- From -500BC to 1976AD if two people wanted to set up a crypto system they had to meet to exchange keys.
- Diffie-Hellman showed a way for two people to establish a shared secret key without meeting- note that all of their communication is public.

Diffie-Hellman

- From -500BC to 1976AD if two people wanted to set up a crypto system they had to meet to exchange keys.
- Diffie-Hellman showed a way for two people to establish a shared secret key without meeting- note that all of their communication is public.
- Hence DH solved a problem that was open for over 2000 years!